UNITED STATES PATENT APPLICATION

FOR

COMPUTER-AIDED DATABASE SYSTEM AND METHOD FOR OPERATING IT

INVENTOR: Andreas Kemmler

Prepared by:
BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN
12400 WILSHIRE BOULEVARD
SEVENTH FLOOR
LOS ANGELES, CALIFORNIA 90025
(408) 720-8598

Attorney's Docket No.   6631P008

"Express Mail" mailing label number: __EV410137018US__   .
Date of Deposit: _____November 21, 2003_____
I hereby certify that I am causing this paper or fee to be deposited with the United
States Postal Service "Express Mail Post Office to Addressee" service on the date
indicated above and that this paper or fee has been addressed to:
Mail Stop Patent Application, Commissioner for Patents, P.O. Box 1450,
Alexandria, VA 22313-1450.

_____Leslie Rogan_____
(Typed or printed name of person mailing paper or fee)

_____
(Signature of person mailing paper or fee)

_____
(Date signed)

COMPUTER-AIDED DATABASE SYSTEM AND METHOD FOR OPERATING IT

[0001] The present patent application claims the priority benefit of the filing date of German Application No. 103 48 665.8 filed October 15, 2003, which is incorporated by reference.

FIELD OF THE INVENTION

[0002] The invention relates to a computer-aided database system and to a method for operating a computer-aided database system.

BACKGROUND OF THE INVENTION

[0003] In a computer-aided database system, such as is used within the framework of SAP R/3, various applications which register, e.g., produce, change or erase, data objects of various data object types in a database associated with the respective application run independently of one another in an application program layer. Such applications are, by way of example, sub-applications of SAP R/3's HR module for personnel, that is to say personnel administration, organization management, personnel development, public sector and the like. To be able to establish which user has registered which data objects at which time, there is provision for record writing. This record writing stores the data object automatically in a database before and after registration, together with further information, for example relating to the registration time and to the user performing the registration. In this context, separate record writing needs to be firmly implemented for each application and each data object type on account of the various data objects and record requirements. The firmly implemented

record writing operations have considerable redundancy in this case, which means that system resources such as memory space are wasted and considerable maintenance complexity also arises. In addition, the firm implementation is inflexible, since it cannot be changed by the database user. If requirements relating to the record writing or data objects are changed or redefined, this involves a high level of complex work. This becomes even greater when the firm record writing operations have a different interface according to data object type and hence need to be called up in different ways. Finally, the application cannot continue to be used during execution of the firmly implemented record writing operation, but rather first needs to wait until the latter has ended.

SUMMARY OF THE INVENTION

[0004] According to one aspect of the present invention, there is provided a computer-aided database system and a method for operating a computer-aided database system in which applications which register data objects of various data object types into a database associated with the respective application can be executed independently of one another in an application layer, where the respectively addressed application in the application layer, in response to a registration operation, produces information about the data object which is to be registered and about the registration, in the form of a reference of prescribed structure, calls up a record writer from a number of record writers (which are implemented in a record layer encapsulated with respect to the application layer) which is associated with the data object type by means of a table and uses a dynamic interface, the record writer accessing the reference and hence the information via the interface and transferring record object data having information about the data object which is to be registered and about the registration to a record registrar for the purpose of permanent storage.

[0005] The transfer of information using a reference to data may allow an interface which is open to all data object types and record writers regardless of their specific implementation. This may mean that the record writers can be used unchanged for various applications and data object types. In this exemplary case, the interface is in dynamic form that is to say is able to access the transferred data differently according to data object type. Together with the table which permits simple changing of the association between a data object type and various record writers, the record writing can easily be changed by the database user. By way of example, the registration of certain data objects can be

recorded no longer as a result of association with a record writer, which does not create any records or can be recorded differently as a result of association with a different record writer. The result is more flexible adaptability while avoiding redundant coding. Since the record layer is additionally encapsulated with respect to the application layer, asymmetric execution of the record writing is made possible. Record writing can therefore work independently of the application, particularly in the background, while the application can continue to be used without restriction. The record layer is encapsulated by a program part, for example an ABAP function block, which is called up by the application layer, performs application-specific transformations on the information and transfers a reference of prescribed structure to the actual record writing.

[0006] The record writers may be in the form of ABAP classes which each contain the dynamic interface and at least one method for further processing of the transferred information. In this context, at least one basic ABAP class is provided. Differing solutions can inherit from this basic ABAP class, can complement, redefine or completely reimplement methods, which results in added adaptability of the record writers to changed requirements, particularly in interaction with the table.

[0007] In addition, a customizing table for further settings by database users may be provided. This expediently allows record writing to be activated or deactivated in detail for particular data object types and possibly sub-types on the basis of differentiation parameters. In this exemplary case, the customizing table may be evaluated before a record writer is called up, so that it is not necessary to call up the record writer for data whose registration is not intended to be recorded.

[0008] The method can be carried out by a computer program, which can be called up from an, in particular, interchangeable data storage medium and/or a data link.

[0009] Further, embodiments and advantages of the invention will become apparent from the following description, drawing and the claims.

BRIEF DESCRIPTION OF THE DRAWINGS

[0010] The invention is explained in more detail below using an exemplary embodiment, which is schematically shown in the appended figure.

DETAILED DESCRIPTION OF AN EXEMPLARY EMBODIMENT

[0011] The database system shown comprises an application layer 1 and a record layer 2, encapsulated with respect to the latter, which are implemented as a computer program in a computer system. By way of example, the record layer 2 is implemented in ABAP-OO and is part of SAP BASIS, i.e. the software layer which forms the basis of almost all SAP applications, the SAP system being able to be used on Windows NT, for example.

[0012] In the application layer 1, applications 3 which register data objects 4 into databases 5 in response to user actions or automatically run independently of one another. In the SAP R/3 HR module, the data objects 4 are the information types.

[0013] The registration is expediently performed using application registrars 6. When a data object 4 is registered, e.g., created, changed or erased, by a database user or a process, the addressed application 3 calls up an associated application registrar 6. The application registrars 6 are program parts which run in the application layer 1 and may be associated with a plurality of applications 3 registering identical data objects 4, and, besides their main task, that of registering data objects 4 in databases 5, are also responsible for transferring information about the registration operation from the application layer 1 to the record layer 2. To this end, an application registrar 6 calls up an encapsulated program part, in this case a function block, which is part of the record writing and conditions the registration for the record writing. This program part produces a reference in line with a standard stipulation, for example in the form

of a pointer to a data structure comprising pointers to the data object 4 before the change, the data object 4 after the change, the change time, the identifier of the user or process initiating the change, etc. One possible data structure is:

HRCDOC_VALUES_REF
        OLD_VALUE_REF   TYPE REF TO DATA
        NEW_VALUE_REF   TYPE REF TO DATA
        CHG_INDICATOR   TYPE CDCHNGINDH

[0014] In this context, a reference to a data object 4 before the change is transferred to OLD_VALUE_REF, and a reference to the data object 4 after the change is transferred to NEW_VALUE_REF, with CHG_INDICATOR indicating the type of registration. When a call is made, a reference to an object or to a structure comprising a plurality of objects can be transferred to OLD_VALUE_REF, for example. The transfer of information is thus open to all data object types regardless of their makeup. Only the internal makeup of the structure HRCDOC_VALUES_REF is binding for all application registrars 6 in this case.

[0015] When the reference has been created, a record writer 7 associated with the data object type can be called up. The associated record writer 7 is ascertained by accessing a table 8, which essentially has a column with data object types and a column with associated record writers 7. In this case, a record writer 7 can be associated with a plurality of data object types, but not the other way round.

[0016] Data object types whose registration does not need to be recorded can either be omitted from the table 8 or can be associated with a record writer 7, which does not create a record.

[0017] In addition, a customizing table for further settings by database users can be provided. In this case, the customizing table is preferably evaluated in the application layer 1 before the table 8 is evaluated for the purpose of assigning a record writer 7. By way of example, the customizing table has the following simplified structure:

MANDT	INFTY	SUBTY	CDOC_ACTIVE
001	1002	*	+
001	1002	1000	+
001	1000	*	-
001	9001	0099	+

[0018] In this context, the column CDOC_ACTIVE indicates for a data identifier MANDT 001, which is used to associate data with a particular area, in this case the client 001, and for various data object types, which in this case are classified using the differentiation parameters Infotype INFTY and Subtype SUBTY by way of example, whether record writing has been activated (+) or deactivated (-), with * being a wildcard for all types. Other differentiation parameters can be used depending on the instance of application.

[0019] In this case, the record writers 7 are implemented in the form of an ABAP class, comprise an interface 9 and at least one method 10 and stipulate for the

respective record object type which records with which information in which form need to be permanently stored.

[0020] The interface 9 is flexible, e.g., no objects or variables but rather, as described above, a reference thereto is transferred. The interface 9 is also dynamic and implements methods for receiving the information from the application layer 1, for ascertaining the data object type associated with the information, for conditioning record object data with information about the data object 4 which is to be registered and about the registration and possibly further data, and for calling up a record registrar 11 for producing and permanently storing a change record on the basis of the record object data, for example in the database 5 corresponding to the data object 4, in a special database or else in the form of a printout or the like.

[0021] The methods 10 implemented in the record writer 7 can contain, besides the methods of the interface 9, methods for setting up parameter tables for the record objects. In this case, each record object has an associated parameter table for data transfer to the record registrar 11, with the call being made generically according to the record object in the form of the ABAP statement CALL FUNCTION func parameter_tables. In this context, the parameter tables need to contain the correct formal parameters for the record registrar 11, which is to be called up. There can also be a method 10 for producing a key which combines fundamental information for the purpose of rapid searching, and expediently a method 10 for checking, whether the record writer 7 is actually responsible for the data object type of the transferred data object 4, so that any misassociation in the table 8 is recognized.

[0022] Expediently, the default solution provided for the record writer 7 is a basic ABAP class that can be used for a large portion of the data object types. The basic ABAP class expediently contains a plurality of different methods for various data object types. From the information transferred in the form of a reference, said basic ABAP class generates corresponding record object data and possibly associated parameter tables and provides methods for calling up a respective suitable record registrar 11.

[0023] On the basis of this basic ABAP class, it is possible to produce specific ABAP classes that take into account peculiarities of a respective data object type. In this case, the specific ABAP classes inherit the interface 9 and the methods 10 from the basic ABAP class, with additional methods 10 being able to be defined and/or the inherited methods 10 being able to be redefined in line with the requirements.

[0024] The application level 1 can also contain a management module 12 for managing and producing the instances of the ABAP classes which form the record writers 6 and possibly for maintaining the table 8 and/or the customizing table. The management module 12 can be used by a user of the database system who has the appropriate authorization. The management module 12 also encapsulates the central methods used by applications, application registrars, record writers etc. by virtue of their being called up, e.g. methods for setting up parameter tables for generically calling up the record registrars 11, methods for recognizing the parameters such as TCODE, UTIME, UDATE etc. which apply as standard to all record registrars 11 and/or record writers 7.

[0025] While the invention has been shown and described with reference to a preferred embodiment, it should be apparent to one of ordinary skill in the art that many changes and modifications may be made without departing from the spirit and scope of the invention as defined in the claims.